Chapter 7
Knowledge Discovery: Temporal Disaggregation in Social Interaction Data

Ritvik Shrivastava and Sreyashi Nag

Abstract This chapter discusses the notion of aggregation over the temporal dimension of temporal digraphs. This issues is particularly important in knowledge discovery and centrality metrics in social networks. We first present an simple temporal adaptation of a random walk based technique (Markov Clustering Algorithm) for community detection and centrality. Later, we present experiments to show the importance of temporal disaggregation in community detection and centrality measures. The experiments show that social structures observed at smaller temporal granularities are in general different from the ones at seen at larger granularities.

Given the ever increasing penetration of communication technologies, it is increasingly becoming feasible to observe and study social interaction data. For instance, it is now possible to answer questions such as "Which communities are transient in nature, i.e., they exist for a short period of time?", "How does an individual memberś social capital (as measured through his/her centrality) fluctuates over time?", etc.

Much of the traditional work [17, 68] in the area of social networks assumed that the underlying social network is does not change with time. However, this is not true in many cases. For example, consider the evolution of an individual's friendship network in an online social networking platforms such as Facebook (www.facebook.com) as he/she moves through different phases professional life (undergraduate, graduate, and full-time job). It is conceivable that strength of some of the ties may go down with time. Similarly, from time to time, we may form some short-lived groups geared towards a specific task, for e.g., a group of faculty getting together for a period of few months over a project proposal. During this time, they may be interacting with each other more frequently than with others. One can find several such real world scenarios, where social aspects of our lives would change with time. As a consequence, it raises the need for developing computational techniques which can consider the time-varying nature of social structures.

To this end, several researchers have started working on this aspect of social interaction data. These works can be broadly classified into following four categories:

- Shortest path centrality metrics for dynamic social networks [28, 30, 40, 41, 61]: These works have extended the traditional shortest path based centrality metrics such as Betweenness [2, 18, 19] and Closeness [19] for temporal digraphs.
- Random walk based centrality for dynamic social networks [47]: This work generalized $\alpha-$centrality for dynamic networks. The key idea of this work was to adapt the random walk such that it gives more weight to interactions which are closer in time. This way it avoids exploring unnecessarily long paths created by stitching interactions over very long periods of time, for example, an interaction between individuals X and Y in Jan 2017 would most likely not be related to an interaction between Y and Z in Jan 2018.
- Persistent community detection in temporal digraphs [5, 62, 64, 65]: These works were one of the first to start considering community detection in a time-varying scenario. The central premise of these works being, each individual would have his "primary" group where he would spend most of his time (i.e., most of his interactions). And other "outside" interactions would not amount to much. Along those lines, the algorithms developed in [5, 62, 64, 65] focused on determining these "primary" groups.
- Transient community detection in temporal networks [33, 46]: Both these works perform a content-based aggregation of individuals into communities. In other words, they can effectively detect communities surrounding a particular news article or topic.

This chapter presents a temporal generalization of a random-walk based technique for community detection on a temporal digraph representation of the social interaction data. This algorithm can discover social structures at both lower temporal granularities (e.g., daily, weekly) as well as higher levels of temporal granularity (e.g., yearly). The key idea in the adaptation is that as a random walk progresses over a temporal digraph (*temporal random walk*), we would first come across transient clusters which would then disperse as the random walk starts considering the network at larger time granularities. In other words, as the random walk progresses over time, only the permanent clusters, i.e., clusters that persist for a significant time period (and in turn, long term centrality metrics for individual nodes), are likely to remain. This technique is capable of detecting transient as well as persistent communities in social interaction data. In addition to community detection, this chapter also presents a random walk based technique for computing a temporal adaptation of the katz centrality [38].

Outline of rest the Chapter: Section 7.1 presents the basic concepts related to temporally detailed social networks (TDSN), transient and persistent communities. In Section 7.2, we discuss the formation of temporal paths in TDSNs. We present a random walk based technique to detect transient communities in Section 7.3. Temporal adaptation of Katz centrality is discussed in Section 7.4. In Section 7.5, we present a case study highlighting the difference between random walk based techniques developed for TDSNs and their traditional counterparts in static graphs.

7.1 Basic Concepts and Problem Definition

A Temporally Detailed Social Network (TDSN) is a collection of panels, where each panel is in turn an aggregation of all the social interactions which happened over a window of fixed length, e.g., minute, hour, day, week, month, etc. Length of this window, hereafter referred to as *TDSN-granularity*, and is decided according to the needs of the application domain. Figure 7.1 illustrates a sample TDSN consisting of 5 panels. Note that if two individuals X and Y interact with each other more than once inside a window, then TDSN would still consist of only one edge between nodes X and Y, however, its weight would be different. Note that it is important to keep the value of *TDSN-granularity* as less as possible. A higher value of this parameter would wrong estimation of number of paths where edges are temporally ordered. Implicitly, we are assuming that all the interactions inside one panel happened at the "same instant".

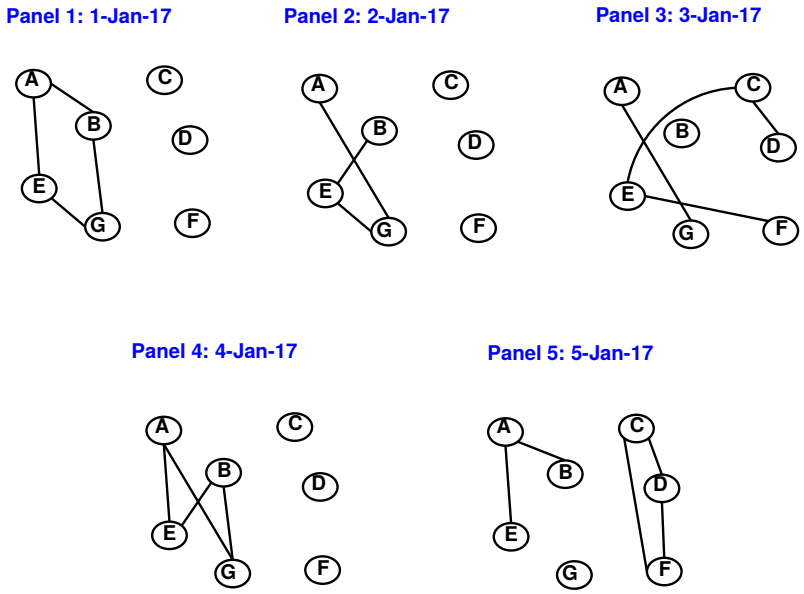


Fig. 7.1 Sample Social Interaction Data

Weight of an edge in a panel: Inside a panel which is an aggregation of all the social interactions that happened over n time units, we compute the weight of an edge P_{ij} between two nodes i and j as follows:

$$P_{ij} = \frac{\#\, times\ `i'\ was\ connected\ to\ `j'}{Length\ of\ the\ panel} \tag{7.1}$$

Communities: Traditionally, a community has been loosely defined as a collection of individuals who happen to be similar to each other in some sense. This could be similarity could arise from having same set of interests, being a part of the same organization, etc. Also, it is generally expected that the individuals would have stronger levels of interaction with people from the same community than with individuals outside the community. In case of temporally detailed social networks, the notion of communities can be further divided into: (a) transient communities and, (b) persistent communities.

Transient Community in a TDSN (C_t) is defined as a group of individuals who interact with each other more often during a short time period. Inside this time interval, these group of individuals interact with each other more often than with others. This communication ceases to exist at other times during in the time horizon of the TDSN. Thus, transient communities have a temporal locality. The members of the transient community need not be in each other's neighborhood and consequently, need not have had any interaction in the past.

Persistent Community in a TDSN (C_p) is defined as a group of individuals who in general interact with each other more than they interact with others. And this level of interaction is maintained throughout the time horizon of the TDSN. Persistent communities have strong interactions between its members by virtue of some level of similarity between them. This similarity is responsible for its prolonged duration during the life span of the network. Transient communities, on the other hand, are possibly created to achieve a certain goal. For example, a group of researchers getting together to write a project proposal.

In a TDSN, random-walk based techniques are based on the concept of temporal paths. The observations for these techniques can be made at variable time granularities. This concept is discussed below.
Temporal Paths : A temporal path P in a TDSN is a sequence of nodes $P = (v_1, v_2, v_3 ... v_n, v_n + 1)$, where each edge in P belongs to a different panel in the TDSN. Moreover, the panel number of (v_i, v_{i+1}) is strictly less than the panel number of (v_{i+1}, v_{i+2}). In other words, a temporal path can be defined as a sequence of temporal edges.

Problem Definition
Input : A collection of panels representing a temporally detailed social network (TDSN).
Output : An algorithm framework for evaluating the presence of transient communities as well as assessing the influence measures of individual nodes.
Objective : Extracting information which can otherwise not be evaluated using traditional methods.

7.2 Formation of Temporal Paths in TDSN

Consider an adjacency matrix M representation of any connected graph, each cell denotes the presence or absence of a direct edge (connection) between each node pair. The cell would contain 1 if there is an direct edge between the pair of nodes,otherwise it would be 0. For such a matrix M, the matrix M^n would contain the number of *n-hop* paths between each pair of nodes. In other words, after multiplying the matrix M n times, each cell (i, j) would contain the number of paths between nodes i and j which have n nodes or less. This is illustrated in Figure 7.2 where *MxM* is a single multiplication which would contain the total number of *0-hop and 1-hop* paths in the network.

M	P	Q	R	S
P	0	1	0	0
Q	0	0	1	0
R	0	1	0	1
S	1	0	0	0

MXM	P	Q	R	S
P	0	0	1	0
Q	0	1	0	1
R	1	0	1	0
S	0	1	0	0

Fig. 7.2 0-hop (i.e., direct connection) and 1-hop path formation.

We now discuss the formation of temporal paths in temporally detailed social networks (TDSN). For sake of simplicity and easy interpretation, in the following discussion, we assume that the weight of every edge in the panels is 1.

Given a collection of panels as a TDSN, we can obtain temporal paths by the multiplication of two or more panels. If $M_1, M_2, \ldots, M_n$ represents the n panels of a TDSN, the resultant matrix $M_t = M_1 \times M_2 \times \ldots \times M_n$ represents the number of temporal paths (refer Section 7.1 for formal definition) of length less than or equal to n, i.e. all temporal paths with at most n intermediate hops between the node pairs. Figure 7.3 illustrates this concept. In this figure, M_1 and M_2 are two consecutive panels in a TDSN. Consider, nodes A and D in the figure. There is no direct connection between them at both snapshots. However, there exists a *1-hop* temporal path A→B→D between them. Note that this process can be easily modified to take into consideration the concept of waiting-time at nodes separately.

M_1	A	B	C	D
A	0	1	0	0
B	0	0	1	1
C	0	1	0	1
D	1	0	0	0

M_2	A	B	C	D
A	0	1	1	0
B	1	0	0	1
C	0	1	0	1
D	1	0	0	0

$M_1 X M_2$	A	B	C	D
A	1	0	0	1
B	1	1	0	1
C	2	0	0	1
D	0	1	1	0

Fig. 7.3 Constructing temporal paths (of length less than or equal to 1) formation in a TDSN.

Considering the weight of edges in panels: Recall the definition of weight of an edge in Section 7.1. One can naturally extend the previously discussed notion of multiplication of panels to consider the weights of the edges. The temporal paths now created would have an implicit notion of strength of connection between two nodes. For instance, consider the panels (belonging to a TDSN) shown in Figure 7.4. In panel M_1 of this network, A was very active in sending emails to B, but not so much with node C. And in panel M_2 of this network, B actively sent emails to D, where C was not so much active in writing emails to E. Thus, when we construct $M_1 \times M_2$ while consider the edge weight, cell corresponding to (A,D) would have a much higher value, implying that the path $A \to B \to D$ was stronger than $A \to C \to E$. Note that, in this example, matrices corresponding to panels were not symmetric. Such aspects depend on the nature of the data available. In case of emails, the matrices would not be symmetric. Whereas, if the dataset contained phone calls, then the matrices corresponding to the panels would have to be symmetric.

M_1	A	B	C	D	E
A	0	0.9	0.2	0	0
B	0	0	0	0	0
C	0	0	0	0.9	0
D	0	0	0	0	0
E	0	0	0	0	0

M_1	A	B	C	D	E
A	0	0	0	0	0
B	0	0	0	0.9	0
C	0	0	0	0	0.2
D	0	0	0	0	0.9
E	0	0	0	0	0

M_1	A	B	C	D	E
A	0	0	0	0.81	0.04
B	0	0	0	0	0
C	0	0	0	0	0.81
D	0	0	0	0	0
E	0	0	0	0	0

Fig. 7.4 Constructing weighted temporal paths (of length less than or equal to 1) in a TDSN.

7.3 Community Identification in TDSNs

The key idea over here is to start a random-walk at any node in the TDSN and continue on to the nodes in the next panel. For instance, if nodes x and y are connected in panel p_i, then the temporal random walk would connect this edges with a out going edge from y, say to node z, in panel p_{i+1}. This makes a temporal random walk $x \to y \to z$ across the panels p_i and p_{i+1}. Such random walks are likely to remain within the same cluster as the paths increase in length.

Given a series of panels of a temporally detailed social network, communities are detected in the following two steps:

Temporal Spreading - In this step, consecutive panels of the TDSN are multiplied sequentially to produce temporal paths. The step produces longer temporal paths with each multiplication. Nodes that are a part of longer paths are likely to be a part of the same community. Since the algorithm uses the weights of the edges while multiplying (instead of just using presence or absence of an edge), this also ensures that only communities with a decent frequency of interaction are discovered. If M_n denotes collection of edges (represented as a matrix) in panel n, w_1 denotes the

lower panel number of the TDSN, and w denotes the number of panels, then result of temporal spreading can be expressed as:

$$TempSpread = \prod_{n=w_1}^{w_1+w} M_n \tag{7.2}$$

Amplification - In this step, the matrix obtained after *temporal spreading* is 'amplified' to strengthen the strong connections and weaken the feeble ones. Here, the value in each cell in the resultant matrix is raised to the αth power (amplification factor). In other words, this step brings the stronger connections into prominence to be identified as clusters/communities and dilutes the weakly connected edges in the network. Thus implicitly, after this step, the algorithms discards noise in the network that may appear as communities themselves.

$$Cell\ (i,j)\ in\ AmplifiedMatrix = \{valueofCell(i,j) in TempSpread\}^{\alpha} \tag{7.3}$$

Following these two steps, communities can be detected in the resultant panel by observing non-zero values in each node row. This procedure is described in the following section.

7.3.1 Approach for the Detection of Transient Communities

The community detection approach described next uses a sliding window (of size ω) to determine communities are different temporal resolutions. The size of the sliding window could vary according to the needs of the social question being investigated. For instance, if the user is interested in uncovering the communities at the temporal resolution of one month; then the length of this sliding window would be number of panels corresponding to 1 month. Similarly, if the user is interested communities which lasted for about 6 months, then the length of this window would be number of panels corresponding to 6 months. The steps of the approach are detailed next:

1. *Step 1: Temporal Spreading* - Given the panels of the TDSN and the sliding window size ω as input, temporal spreading is carried out for all the panels which fall in the window as it slides over all the panels of the input temporally detailed social network.
2. *Step 2: Amplification* - After each instance of temporal spreading, amplification is carried out according to the given amplification factor α.
3. *Step 3: Cluster Extraction* - Following each amplification, all the likely clusters that have been detected at that stage are extracted out and stored as *transient communities*. This is done by extracting for every node row, the nodes with non-zero values. One may also choose to have to some thresholds while choosing the nodes. A very low value in the row may not be an indicator of strong cluster.

After Step 3, one can increase the sliding window size ω and repeat steps 1-3 to get transient communities at larger temporal resolutions.

Optional step 4: Communities at Persistent Level - This can determined when ω is set to the total number of panels in the given TDSN. At this stage, the communities that are extracted using steps 1 to 3 can be stored as *persistent communities*. Algorithm 8 presents the steps involved in the detection of the set of transient and persistent communities present in the input TDSN.

Algorithm 8 Community Detection in Temporally Detailed Social Networks

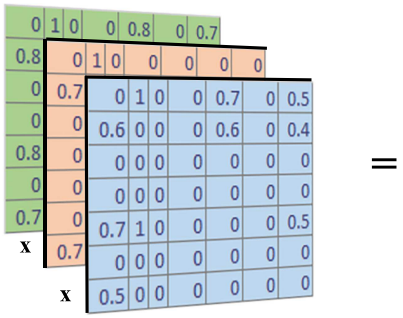
1: $\alpha \leftarrow$ Amplification Factor
2: $w_{lower} = 1$ /* w_{lower} is set to the first panel in TDSN */
3: **while** $\omega \leq$ number of panels in TDSN **do**
4: **while** $w_{lower} + \omega \leq$ last panel of TDSN **do**
5: $TempSpread = TemporalSpread(w_{lower}, \omega)$ /*Multiplies panels w_{lower} through $w_{lower+\omega}$*/
6: $Amp = Amplify(TempSpread, \alpha)$
7: Extract communities from Amp
8: $w_{lower} = w_{lower} + 1$ /* Window is slid by 1 panel */
9: **end while**
10: $\omega = \omega + 1$ /* Increase the resolution of interest by increasing the length of the sliding window */
11: **end while**
12: Return all transient and persistent communities

7.3.2 Running Example for Transient Community Detection

Figure 7.5 illustrates the discussed algorithm on a sample synthetic dataset. This dataset consists of a 7-node TDSN over a time period of 200 timestamps. The snapshots were aggregated into 20 panels, each consisting of 10 snapshots. Weights of the edges inside the panel were computed using the equation discussed in Section 7.1. Steps 1 to 4 (optional) are shown in Figure.

7.4 Node Influence Measures in TDSNs

In this section, we generalize Katz Centrality for temporally detailed social networks. This would help in studying the "influence" of nodes in the TDSN.



	A	B	C	D	E	F	G
A	0.06	0.28	0.13	0	0.24	0	0.25
B	0.14	0.32	0.15	0	0.43	0	0.52
C	0.13	0.22	0	0	0.47	0	0.3
D	0.1	0.16	0	0	0.35	0	0.23
E	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0
G	0.16	0.53	0.25	0	0.47	0	0.58

Step 1: Temporal Spreading with factor = 3

	A	B	C	D	E	F
A	0	0.08	0.02	0	0.06	0
B	0	0.1	0.02	0	0.18	0
C	0	0.05	0	0	0.22	0
D	0	0.03	0	0	0.12	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

Step 2: Result of Amplification with factor = 2

	A	B	C	D	E	F	G
A	0.69	1.29	0	0	1.14	0	1.18
B	0.69	1.29	0	0	1.14	0	1.18
C	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0
E	1.21	2.26	0	0	2	0	2.06
F	0	0	0	0	0	0	0
G	1.47	2.75	0	0	2.43	0	2.5

Step 4: Result at Persistent Level

	A	B	C	D	E	F	G
B	0	0.1	0.02	0	0.18	0	0.27

Step 3: Cluster Extraction corresponding to node row B. Nodes that are part of the extracted cluster have been highlighted.

Fig. 7.5 Running Example of Community Detection in a TDSN

7.4.1 Temporal Katz Centrality

Traditional Katz Centrality measures the total number of direct and indirect paths of all lengths incident on a particular node [38]. This is done while attenuating the effect of longer paths. Following is a mathematical definition of the Katz centrality on static graphs.

$$x_i = \sum_{k=1}^{\infty} \sum_{j=1}^{N} \gamma^k (A^k)_{ji} \tag{7.4}$$

Here, γ is the attenuation factor between 0 and 1, and N is the number of nodes in the input graph. This centrality measure indicates the relative influence of a node in the network.

We now present an temporal adaption of this metric for TDSNs. The adaptation operates under the assumption that as the length of the paths increases, the impact

of the information traveling through it is reduced. This is taken into account when observing the number of viable temporal paths ending at the node under consideration. In order to account for this, the algorithm attenuates the contribution of longer temporal paths, similar to what the traditional Katz Centrality does in static graphs.

Algorithm 9 presents a pseudocode to compute a temporal adaptation of the Katz centrality on TDSNs. The algorithm takes the following three inputs: (a) initial panel t_i, (b) last panel to consider t_f, (c) window size ω. By choosing setting appropriate values to these parameters, the algorithm can answer a wide variety of these questions. Following are few examples:

1. *Case 1:* In how many different ways can information reach a node through temporal paths of length 1 day between May 1 and August 1? For answering this question: $t_i = 1^{st}$ panel corresponding to May 1, $t_f =$ last panel corresponding to August 1, $\omega = 1$ *day* and the attenuation factor γ should be set to 1.
2. *Case 2:* In how many different ways can information reach a node through temporal paths of length 1 day for the entire duration of the network's existence? For answering this question: $t_i = 1^{st}$ panel in TDSN, $t_f =$ last panel in TDSN, $\omega = 1$ *day* and the attenuation factor γ should be set to 1.
3. *Case 3:* In how many different ways can information reach a node through temporal paths of all lengths between January 1 and February 1? For answering this question: $t_i = 1^{st}$ panel corresponding to Jan 1, $t_f =$ last panel corresponding to Feb 1, $\gamma = 1$ $\omega = 1-day, 2-days, 3\ days, \ldots,$ time horizon of the input TDSN.
4. *Case 4:* In how many different ways can information reach a node through temporal paths of all lengths for the entire duration of the network's existence? For answering this question: $t_i = 1^{st}$ panel in TDSN, $t_f =$ last panel in TDSN, $\gamma = 1$ $\omega = 1-day, 2-days, 3\ days, \ldots,$ time horizon of the input TDSN.

Algorithm 9 presents a procedure for calculating temporal katz centrality for cases 3 and 4 only. Cases 1 and 2 can be computed in a straightforward fashion by setting the value of ω appropriately instead of incrementing it in each iteration as done by Algorithm 9.

Algorithm 9 Katz Centrality for Temporally Detailed Social Networks

$t_i \leftarrow$ panel in TDSN from where analysis starts /*Given as input */
$t_f \leftarrow$ panel in TDSN from where analysis stops /*Given as input */
$\gamma \leftarrow$ Attenuation Factor /*Given as input */
$\omega = 1$ /*Length of the sliding window, incremented after each iteration */
$w_{lower} = t_i$
while $\omega \leq t_f - t_i$ **do**
  while $w_{lower} + \omega \leq t_f$ **do**
    $M = TemporalSpread(w_{lower}, \omega)$ /*Multiplies panels w_{lower} through $w_{lower+\omega}$*/
    $K_i = \gamma^{\omega} \sum_{j=1}^{n} M_{i,j}$ /* n is the #distinct nodes in TDSN*/
    $K = K + K_i$
    $w_{lower} = w_{lower} + 1$
  end while
  $\omega = \omega + 1$
end while

7.5 Case Study on TDSN

In this section, we present a case study which illustrates the key nature of temporally detailed social networks. More specifically, this case study highlights the fact that social structures (community and social capital) observed in smaller temporal resolutions (e.g., month) are different from the ones seen in larger resolutions (e.g., years). Following two datasets are used in the case study:

1. **University Email Dataset:** This dataset contains email communications from a large European university. This dataset recorded about 161227 email communications among 4554 individuals. It has been used for multiple studies previously [28, 49, 50]. These email communications were recorded in the following format: $< A, B, t >$, where A and B are two individuals in the University and t denotes the time-stamp (in minutes) at which individual A sent an email to individual B.
2. **CollegeMsg Dataset [58]:** This dataset comprises of private messages sent on an online social network at the University of California, Irvine over a duration of 193 days. Users could search the network for others and then initiate conversation based on profile information, hence creating an edge $< u, v, t >$, where user u sent a private message to user v at time t.

7.5.1 Metric used for Experimentation

Jaccard Index: The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity of sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the cardinality of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{7.5}$$

A lower value of the Jaccard index implies that the sets A and B (or communities in this case) are dissimilar from each other. In our case, a lower Jaccard score while comparing persistent communities to communities at smaller temporal resolution signifies the presence of a transient groups. In this case study, the Jaccard Index is obtained for each node in the network by comparing every transient group discovered at each time granularity with the persistent community obtained at the persistent level. For example, consider a network of 6 nodes A, B, C, D, E, F. Assume that at the level of persistent communities nodes A, B and E form a community. Whereas, at some intermediate temporal resolutions (i.e., at lower values of ω), assume that nodes A, C and F form a community as per our algorithm. If the set of nodes corresponding to the persistent community of node A is set P_A, and the transient ones is set T_A. We calculate Jaccard(P_A,T_A) as $\frac{sizeOf(P_A \cap T_A)}{sizeOf(P_A \cup T_A)}$. The set $P_A \cup T_A$ consists of nodes A,B,C,E,F, whereas the set $P_A \cap T_A$ consists of only node A. The

resultant Jaccard(P_A,T_A) value for node A would be 1/5 = 0.2. This is represented through a confusion matrix in Figure 7.6.

Both in Transient and Persistent	In Transient but Not in Persistent
Not in Transient but in Persistent	In Neither (Irrelevant Nodes)

1	2
2	Irrelevant

$$\text{Jaccard}(P_A,T_A) = \frac{sizeOf(P_A \cap T_A)}{sizeOf(P_A \cup T_A)} = \frac{1}{1+2+2} = \frac{1}{5} = 0.2$$

Fig. 7.6 Confusion Matrix Representation of Jaccard Index.

7.5.2 Experiment 1

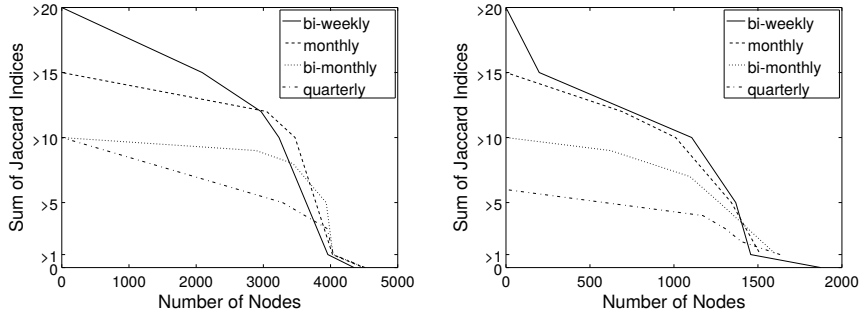
Hypothesis - Community structures present at smaller temporal resolutions are different than the ones present at the final level–where ω is set to entire time horizon of the input TDSN.
Experimentation - In order to investigate the above hypothesis, we compared the communities detected for different values of temporal resolutions (i.e., length of the sliding window ω) against the ones detected at the persistent level, i.e. the communities detected while viewing the entire TDSN (ω = entire time horizon of the input TDSN). We used the previously discussed Jaccard index to compare these. For a particular value of ω and a particular individual, as the sliding window moves forward, we add the resultant Jaccard scores across all positions of the sliding window in the TDSN to obtain a sum. If T_i is a transient community for node A at one position of the sliding window and P be its persistent community found at top level; then the sum of Jaccard indices for node A across θ positions of the sliding window (for same ω), is calculated using:

$$Sum_i = \sum_{i=1}^{\theta} J(T_i, P) \tag{7.6}$$

A smaller sum indicates a larger difference between the communities detected at lower time granularities when compared to those detected at the persistent level. In other words, a smaller sum value signifies a greater proportion of transient communities detected at that granularity. Figure 7.7 illustrates the results of this experiment. The Figure illustrates that, for a large number of individuals (nodes in the Figure), there is a huge difference between their transient and their persistent communities.

7.5.3 Experiment 2

Hypothesis - Information obtained from TDSNs (which implicitly preserve the temporal sequence of interaction events) cannot be obtained by static analysis per-



(a) University Email dataset (b) CollegeMsg dataset

Fig. 7.7 Sum of Jaccard indices calculated for different values of ω for both datasets.

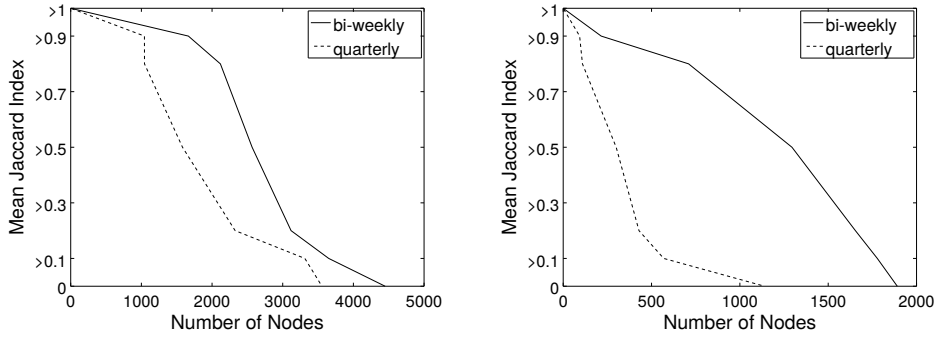
formed on the aggregated TDSN.

Experimentation - To investigate this hypothesis, we compare the communities detected by static analysis on aggregated TDSNs against those detected by our temporal random-walk based approach. For every size of the sliding window (ω), we extracted communities using the approach discussed in this chapter. Following this we created a static graphs by aggregating (collapsing the time dimension) all the panels for every distinct position of the sliding window, as it slides along the panels in the TDSN. On these static graphs, we applied the traditional Markov Clustering Algorithm to extract communities. Using the Jaccard index, we evaluated the difference between the two sets of communities. We calculate the mean Jaccard index for a node across a particular value of ω as a measure of this difference.

Figure 7.8 shows the results of this experiment. One can observe that, in both the CollegeMsg and the University Email datasetes, mean Jaccard index for a window length of one quarter is less than that for two weeks. This is because of the fact that in case of one quarter, we would be doing more aggregation and thus inflicting more damage to the temporal ordering of interactions. As a result, the communities obtained from aggregated panels would be much more different than the ones seen while respecting the temporal ordering of interactions (as done in the algorithm presented in this chapter). This wide difference shows up as low values of mean Jaccard Index for several nodes in the network.

7.5.4 Experiment 3

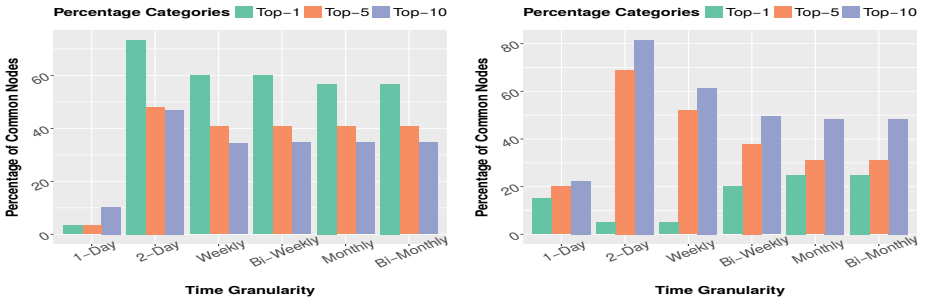
Hypothesis - The influence of nodes measured at distinct temporal resolutions is different from their influence in the aggregated representation of social interactions.



(a) University Email dataset (b) CollegeMsg dataset

Fig. 7.8 Mean of Jaccard indices for comparing time-aware approach against the MCL algorithm over different temporal resolutions. ω was set to 2 weeks and 1 quarter.

Experimentation - In this experiment we compare the result of Temporal Katz centrality presented in this chapter against the traditional Katz Centrality method applied on the aggregated data. For the purpose of presentation, we chose to study Case 3 of the proposed Temporal Katz Centrality as described in Section 7.4. The initial time was set to the beginning of the TDSN and the final time as the 2-month ahead of the the initial time. The proposed algorithm was run on every window size, between 1 day and 2 months. At each window size the top 1%, 5% and 10% most influential nodes were recorded. This data at every window size was compared against the the top 1%, 5% and 10% most influential nodes obtained through the traditional Katz Centrality approach applied on the static graph obtained by aggregating the panels of the window.



(a) University Email dataset (b) CollegeMsg Dataset

Fig. 7.9 Traditional vs temporal Katz Centrality

Figure 7.9 presents the results of this experiment. As the figure shows, with exception of window length of 1 day, in general there is decrease in agreement as the window length is increased from 2 days to 2 months. This is again intuitive because as we increase the length of window, in case of traditional katz, we would be doing

more aggregation and thus inflicting more damage to the temporal ordering of interactions. This results in greater disagreement (thus lower values of Jaccard index) between temporal Katz (computed on TDSN) and traditional Katz (computed on aggregation of panels).

7.6 Conclusion

Using the approach presented in this chapter, we observed that the communities seen at smaller time granularities are quite different than the ones seen at larger granularities. We also showed that information obtained by detailed temporal analysis of TDSNs is vastly different from that attained from aggregated versions of the same graph. Lastly, our modified katz centrality calculated a member's influence on the network at smaller time granularities which was seen to differ from the static metric applied at the aggregated scale.

References

1. Ali, R.Y., Gunturi, V.M., Kotz, A.J., Shekhar, S., Northrop, W.F.: Discovering Non-compliant Window Co-Occurrence Patterns: A Summary of Results, pp. 391–410 (2015)
2. Anthonisse, J.M.: The rush in a directed graph. CWI Technical Report Stichting Mathematisch Centrum. Mathematische Besliskunde-BN 9/71, Stichting Mathematisch Centrum (1971)
3. Batchelor, G.: An introduction to fluid dynamics. Cambridge University Press (1973)
4. Berge, C.: Graphs and Hypergraphs. Elsevier Science Ltd (1985)
5. Berger-Wolf, T.Y., Saia, J.: A framework for analysis of dynamic social networks. In: Proc. of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pp. 523–528. ACM (2006)
6. Braha, D., Bar-Yam, Y.: From centrality to temporary fame: Dynamic centrality in complex networks. Complexity **12**(2), 59–63 (2006)
7. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology **25**(2), 163–177 (2001)
8. Chabini, I.: Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. Transportation Research Record: Journal of the Transportation Research Board **1645**(-1), 170–175 (1998)
9. Chabini, I., Lan, S.: Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. IEEE Transactions on Intelligent Transportation Systems **3**(1), 60–74 (2002)
10. Dehne, F., Omran, M.T., Sack, J.R.: Shortest paths in time-dependent FIFO networks using edge load forecasts. In: Proceedings of the Second International Workshop on Computational Transportation Science, IWCTS '09, pp. 1–6 (2009)
11. Delling, D., Nannicini, G.: Bidirectional core-based routing in dynamic time-dependent road networks. In: Algorithms and Computation, pp. 812–823 (2008)
12. Demiryurek, U., Banaei-Kashani, F., Shahabi, C.: A case for time-dependent shortest path computation in spatial networks. In: Proc. of the ACM SIGSPATIAL Intl. Conf. on Advances in GIS, GIS '10, pp. 474–477 (2010)
13. Demiryurek, U., Banaei-Kashani, F., Shahabi, C., Ranganathan, A.: Online computation of fastest path in time-dependent spatial networks. Advances in Spatial and Temporal Databases pp. 92–111 (2011). Springer. LNCS 6849
14. Demiryurek, U., Banaei-Kashani, F., Shahabi, C., Ranganathan, A.: Online computation of fastest path in time-dependent spatial networks. In: Proc. of the 12th intl. conf. on Advances in spatial and temporal databases, SSTD'11, pp. 92–111. Springer-Verlag (2011)
15. Ding, B., Yu, J., Qin, L.: Finding time-dependent shortest paths over large graphs. In: Proceedings of the 11th international conference on Extending database technology: Advances in database technology, pp. 205–216. ACM (2008)
16. Eckmann, J.P., Moses, E., Sergi, D.: Entropy of dialogues creates coherent structures in e-mail traffic. Proceedings of National Academy of Sciences **101**(40), 143,330–14,337 (2004)
17. (Editor), C.C.A.: Social Network Data Analytics. Springer (2011)
18. Freeman, L.: Centrality in social networks conceptual clarification. Social networks **1**(3), 215–239 (1979)
19. Freeman, L.C.: A Set of Measures of Centrality Based on Betweenness. Sociometry **40**(1), 35–41 (1977)
20. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. Elsevier, Discrete applied mathematics **42**(2), 177–201 (1993)
21. George, B., Kim, S.: Spatio-Temporal Networks. Springer, New York (2003). DOI 10.1007/978-1-4614-4918-8
22. George, B., Kim, S., Shekhar, S.: Spatio-temporal network databases and routing algorithms: a summary of results. In: Proceedings of the 10th international conference on Advances in spatial and temporal databases, SSTD'07, pp. 460–477. Springer-Verlag, Berlin, Heidelberg (2007)

23. George, B., Shekhar, S.: Time-aggregated graphs for modeling spatio-temporal networks. Advances in Conceptual Modeling-Theory and Practice pp. 85–99 (2006). Springer
24. George, B., Shekhar, S., Kim, S.: Spatio-temporal network databases and routing algorithms. Tech. Rep. 08-039, University of Minnesota - Computer Science and Engineering (2008)
25. Gunturi, V., Shekhar, S., Bhattacharya, A.: Minimum spanning tree on spatio-temporal networks. In: Proc. of the 21st Intl. Conf. on Database and expert systems applications: Part II, DEXA'10, pp. 149–158 (2010)
26. Gunturi, V.M.V., Nunes, E., Yang, K., Shekhar, S.: A critical-time-point approach to all-start-time lagrangian shortest paths: A summary of results. Advances in Spatial and Temporal Databases pp. 74–91 (2011). Springer. LNCS 6849
27. Gunturi, V.M.V., Shekhar, S.: Lagrangian Xgraphs: A Logical Data-Model for Spatio-Temporal Network Data: A Summary, pp. 201–211. Springer (2014)
28. Gunturi, V.M.V., Shekhar, S., Joseph, K., Carley, K.M.: Scalable computational techniques for centrality metrics on temporally detailed social network. Machine Learning (2016). DOI 10.1007/s10994-016-5583-7
29. Gunturi, V.M.V., Shekhar, S., Yang, K.: A critical-time-point approach to all-departure-time lagrangian shortest paths. IEEE Transactions on Knowledge and Data Engineering **27**(10), 2591–2603 (2015)
30. Habiba, Tantipathananandh, C., Y. Berger-Wolf, T.: Betweenness centrality measure in dynamic networks. Tech. Rep. 2007-19, Center for Discrete Mathematics and Theoretical Computer Science (2007)
31. Howison, J., Wiggins, A., Crowston, K.: Validity issues in the use of social network analysis with digital trace data. Journal of the Association for Information Systems **12**(12) (2011)
32. Jing, N., Huang, Y.W., Rundensteiner, E.A.: Hierarchical optimization of optimal path finding for transportation applications. In: Proceedings of the fifth international conference on Information and knowledge management (CIKM), pp. 261–268 (1996). ACM
33. Kamath, K.Y., Caverlee, J.: Transient crowd discovery on the real-time social web. In: Proc. of ICWSM, pp. 585–594. ACM (2011)
34. Kanoulas, E., Du, Y., Xia, T., Zhang, D.: Finding fastest paths on a road network with speed patterns. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE), p. 10 (2006). IEEE
35. Karagiannis, T., Vojnovic, M.: Email information flow in large-scale enterprises. Technical Report Microsoft Research (2008)
36. Kargupta, H., Gama, J., Fan, W.: The next generation of transportation systems, greenhouse emissions, and data mining. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1209–1212. ACM (2010)
37. Kargupta, H., Puttagunta, V., Klein, M., Sarkar, K.: On-board vehicle data stream monitoring using minefleet and fast resource constrained monitoring of correlation matrices. New Generation Computing **25**(1), 5–32 (2006)
38. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
39. Kaufman, D.E., Smith, R.L.: Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. I V H S Journal **1**(1), 1–11 (1993)
40. Kim, H., Anderson, R.: Temporal node centrality in complex networks. Physcial Review E **85**(2) (2012)
41. Kim, H., Tang, J., Anderson, R., Mascolo, C.: Centrality prediction in dynamic human contact networks. Computer Networks **56**(3), 983 – 996 (2012)
42. Kleinberg, J., Tardos, E.: Algorithm Design. Pearson Education (2009)
43. Köhler, E., Langkau, K., Skutella, M.: Time-expanded graphs for flow-dependent transit times. In: Proceedings of the 10th Annual European Symposium on Algorithms, ESA '02, pp. 599–611. Springer-Verlag, London, UK, UK (2002)
44. Kossinets1, G., Watts, D.J.: Empirical analysis of an evolving social network. Science **311**(5757) (2006)
45. Kulldorff, M.: A spatial scan statistic. Communications in Statistics - Theory and Methods **26**(6), 1481–1496 (1997)

46. Lehmann, J., et al.: Transient news crowds in social media. In: ICWSM (2013)
47. Lerman, K., Ghosh, R., Kang, J.h.: Centrality metric for dynamic network analysis. In: Proceedings of KDD workshop on Mining and Learning with Graphs (MLG) (2010)
48. Liu, H., Hu, H.: Smart-signal phase ii: Arterial offset optimization using archived high-resolution traffic signal data. Tech. Rep. CTS 13-19, Intel. Trans. Sys. Inst., Center for Transportation Studies, Univ. of Minnesota (Apr-2013)
49. Malmgren, R.D., et al.: A poissonian explanation for heavy tails in e-mail communication. Proc. of the National Academy of Sciences **105**(47), 18,153–18,158 (2008)
50. Malmgren, R.D., et al.: Characterizing individual communication patterns. In: Proc. of the 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, KDD '09, pp. 607–616 (2009)
51. Nannicini, G.: Point-to-point shortest paths on dynamic time-dependent road networks. 4OR **8**(3), 327–330 (2010)
52. Nannicini, G., Delling, D., Liberti, L., Schultes, D.: Bidirectional a* search for time-dependent fast paths. In: Experimental Algorithms, pp. 334–346. Springer Berlin Heidelberg (2008)
53. Nannicini, G., Delling, D., Schultes, D., Liberti, L.: Bidirectional a* search on time-dependent road networks. Networks **59**(2), 240–251 (2012)
54. NAVTEQ: `www.navteq.com`
55. Neill, D.B., Moore, A.W.: Rapid detection of significant spatial clusters. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 256–265. ACM (2004)
56. Nia, R., Bird, C., Devanbu, P., Filkov, V.: Validity of network analyses in open source projects. In: 2010 7th IEEE Working Conference on Mining Software Repositories (MSR), pp. 201–209 (2010)
57. Orda, A., Rom, R.: Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. Journal of the ACM (JACM) **37**(3), 607–625 (1990). ACM
58. Panzarasa, P., et al.: Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. Journal of the American Society for Information Science and Technology **60**(5), 911–932 (2009)
59. Shekhar, S., Gunturi, V., Evans, M.R., Yang, K.: Spatial big-data challenges intersecting mobility and cloud computing. In: MobiDE, pp. 1–6. ACM (2012)
60. Shekhar, S., Liu, D.: Ccam: A connectivity-clustered access method for networks and network computations. Knowledge and Data Engineering, IEEE Transactions on **9**(1), 102–119 (1997)
61. Tang, J., Musolesi, M., Mascolo, C., Latora, V., Nicosia, V.: Analysing information flows and key mediators through temporal centrality metrics. In: Proceedings of the 3rd Workshop on Social Network Systems, SNS '10, pp. 3:1–3:6 (2010)
62. Tantipathananandh, C., Berger-Wolf, T.: Constant-factor approximation algorithms for identifying dynamic communities. In: Proc. of the 15th ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining, pp. 827–836. ACM (2009)
63. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, pp. 717–726. ACM, New York, NY, USA (2007)
64. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proc. of the 13th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data mining, pp. 717–726. ACM (2007)
65. Tantipathananandh, C., Berger-Wolf, T.Y.: Finding communities in dynamic social networks. In: Data Mining (ICDM), 2011 IEEE 11th Intl. Conf. on, pp. 1236–1241
66. Tyler, J.R., Tang, J.C.: When can i expect an email response? a study of rhythms in email usage. In: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work, pp. 239–258 (2003)
67. Wu, F., Huberman, B.A.: Novelty and collective attention. Proc. Natl. Acad. Sci. USA **104**(45), 17,599–17,601 (2007)
68. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Comput. Surv. **45**(4), 43:1–43:35 (2013)

69. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: Proc. of the SIGSPATIAL Intl. Conf. on Advances in GIS, GIS '10, pp. 99–108 (2010)
70. Zheng, Y., Zhou, X.E. (eds.): Computing with Spatial Trajectories. Springer (2011)